ROBOTIC ARM CONTROLLER USING COMPUTER INTERFACING

Mohamad Yusri Bin Ramliy
Faculty of Electrical Engineering
Universiti Teknologi MARA Shah Alam
40450 Selangor Malaysia
Email: gunner_aku@yahoo.com

ABSTRACT

This project is about to develop a robotic arm controller by using computer interfacing. The technique uses the capabilities of the computer's parallel port to provide an 8-bit input. The software that is use to provides the interface is Microsoft Visual Basic. The scope of this project is to control the robotic arm to pick the specified object and move it to the specified angle or direction At the end of this project, the robotic arm finally can be operated by itself automatically at actual job just by clicking on one switch or button. In this project, ROB 3 – EUROBTEC was used to interface with the programming.

Keywords: Visual Basic, Computer Interfacing, Robotic Arm, Parallel Port.

1.0 INTRODUCTION

As technology progressed and the need for greater external connectivity increased, the parallel port becomes the means by which users could connect higher performance peripherals. In this project, parallel port will be use as a medium to interface between the software and the robotic arm. The reason of using parallel port in this project is its maximum data transfer rate, which is around 150 kilobytes per seconds, and it is extremely software intensive. However, as great as this parallel port itself has, it also has limitation which is the connection can be carried out not more than 6 feet using the external cables [4].
This project is concern in controlling the robotic arm by using computer interfacing and visual basic software. The important parts of the programming for this project are:

1. Accuracy and time actual to do some job
2. Choose a right motor to move a part of robot.

1.1 OBJECTIVE

The objective for this project is to develop a programming system that can train a robotic arm as well as its can do the job at actual environment. The main objectives of this project are:

1. To developed an interactive application software using Microsoft Visual Basic that will control the output of the parallel port.
2. To be able to interface the robotic arm with the computer by using interfacing circuit.
3. To achieve the given task, which is picking an object, move the object to the given location and return to original position.

1.2 MICROSOFT VISUAL BASIC 6.0

Visual Basic [VB] is one of the most exciting developments in programming in many years. A programming language uses visual techniques to illustrate the relationship between data or the changes to the data. Visual Basic 6.0 requires the Microsoft Windows operating system to operate [1].

1.3 INTERFACING

This project consists of four major parts, which are the PC as the master part, the feedback circuitry, the interface circuitry and a robotic arm. Figure 1 shows the interfacing for this project that describes the whole system integration and operation. The PC will make connection to the other major parts using a bidirectional parallel peripheral interface (parallel port). Again, parallel port is used because it has bidirectional support and many input and output in one port. In this project, four inputs and eight outputs will be use to connect or interface within the circuit [4].

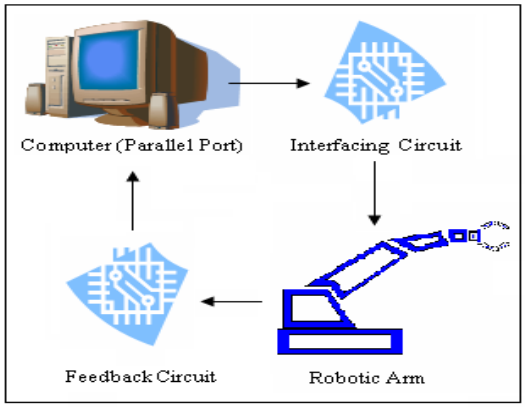

Figure 1: Interfacing of the Parallel Port

2.0 SCOPE OF WORK

1) Study the visual basic program.
2) Write a Visual Basic programming language to interface with the interfacing circuit and robotic arm.
3) Collect data and modified the program.

3.0 METHODOLOGY

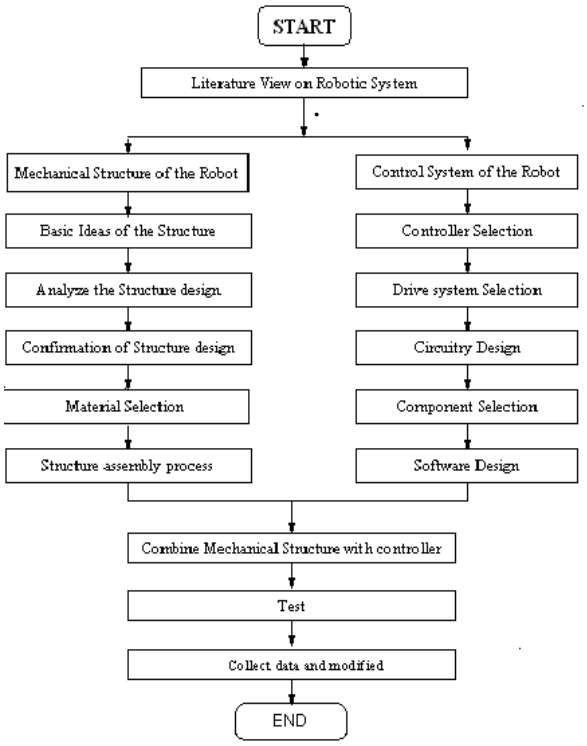

Figure 2: The Process Flow of This Project

Referring to Figure 2, there are 4 main things that have to be considered before making this project successfully. First thing is to understand about the movement of the robotic arm itself. Secondly, know and study about the controlling system for the robotic arm. Then, know on how to interface that robotic arm with both parallel port and software and finally, testing both hardware and software.

3.1 SOFTWARE IMPLEMENTATION

Visual basic is an object oriented language function. This software is one of the most flexible, powerful language and easy to read and write [1]. Actually, this software simplifies the Windows programming. Visual Basic 6.0 consists of two types of object, which are called forms and controls [2]. We can see these objects in Visual Basic start up. The form is used to design the user interface for a Visual Basic application. Same with other programming such C programming, C++ and JAVA, it has if/else/then, selects case, do/loop until and arrays instruction. Visual Basic 6.0 is used to send the data to the hardware through the parallel port. The movement and data will be displayed according to the user instruction. Furthermore, User can easily use this software to monitor a system without any complication in understanding its functions.

By using Visual Basic, direct port controlling from application is not possible if we use Windows XP and to be able to control the parallel port directly you will need to write some kind of device driver into the system. Therefore, to interface the computer parallel port with VB, user needs to install io.dll and inpout.dll files into their system. Io.dll provides a useful set of commands for reading and writing to the I/O ports. These commands are consistent between 95/98 and NT/2000/XP. Furthermore, there is no need for the programmer to learn assembly language or muck with kernel mode drivers. Simply link to the DLL and call the functions. Then, some declaration must be included in the VB programming.

For this project, when the user pushes the start button, the robotic arm will load an automatic program. Then, the program will command the robotic arm to pick up an object and move it to the different location or angle. The robotic arm will run automatically until it moves the specified object and put it onto another place. After that, the robotic arm will move back to its initial condition. Upon completing the task that

is given, the robotic arm has to make a few steps of moves in order to move the object systematically. Note that timer for VB is in millisecond. Therefore, all value will be multiplied by 1000 in the program to get the exact time.

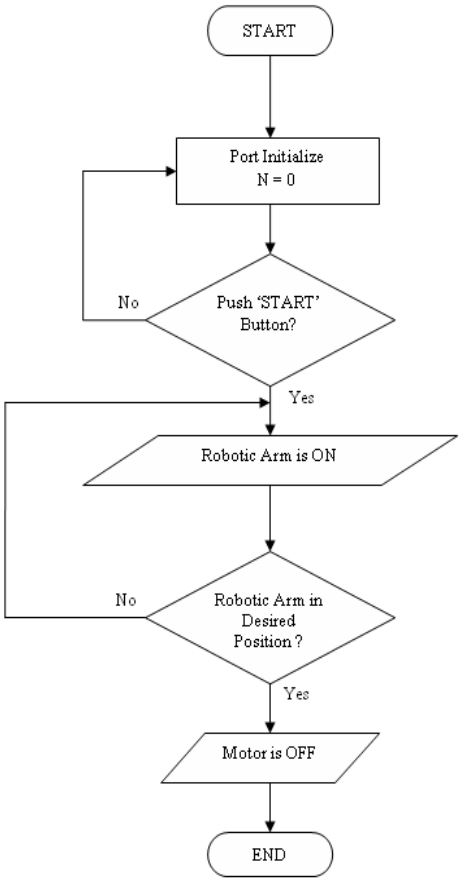

Figure 3: Flow Chart for the Whole System Software

3.2 HARDWARE DEVELOPMENT

The structure of robot contains 6Vdc motor and DC servomotor. The gripper, wrist and tool are using servomotor while the shoulder, elbow and base are using 6Vdc motor. Each motor is rotate at two directions, either clockwise or anti-clockwise. Table 1 showing short form of difference rotation direction at each motor. Meanwhile, the movement of the shoulder and base will be measured using an angular switch. The contact switch has equally space slots and it represent the interval movement of angle. Output signal "1" or "0" will be produced when a shaft of angular switch is rotating. Finally, the movement of the elbow will be measured using the variable resistor. The signal that produced from the variable resistor will be send to the ADC circuit and it will measure the analog signal from variable resistor, and then converted the analogue signal into the digital signal with 8 bit.

	Motor	Rotation of Motor	
		Clockwise	Anti - Clockwise
1	Base	Axis 1mod13	Axis 1mod14
2	Shoulder	Axis 2mod11	Axis 2mod12
3	Elbow	Axis 3mod9	Axis 3mod10
4	Wrist	Axis 4mod7	Axis 4mod8
5	Tool	Axis 5mod5	Axis 5mod6
6	Gripper	Axis 6mod3	Axis 6mod4

Table 1: Rotational Direction at Each Motor.

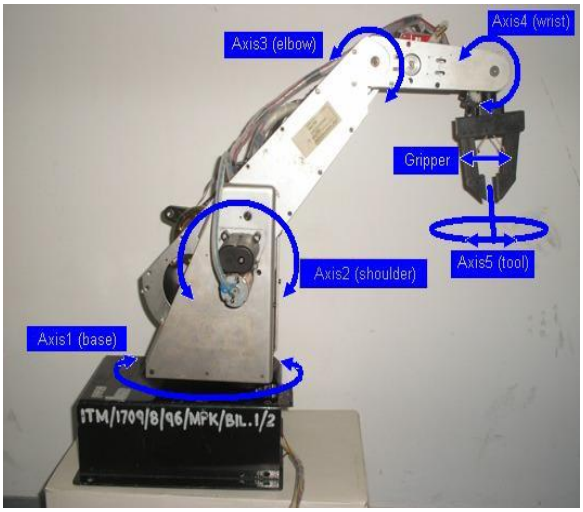

Figure 4: The Movement of Axis at Each Motor

To solve limited output from parallel port, a counter and 4 to 16 decoder circuits were then are designed to extend the number of outputs. It will use pulse signal from the programming or software with certain counting and from the pulse given, the counter and 4 to 16 decoder will select and operate the desired motor. Refer to the Table 2 and Figure 5, the impulse signal will be sent to the input of the counter (IC 74293). This counter will then send signals to the IC 4514 and this IC will allow a selected output to remain high and send it to the base of the transistor. The transistor will amplify the signal to activate a relay and causing a motor to operate (ON).

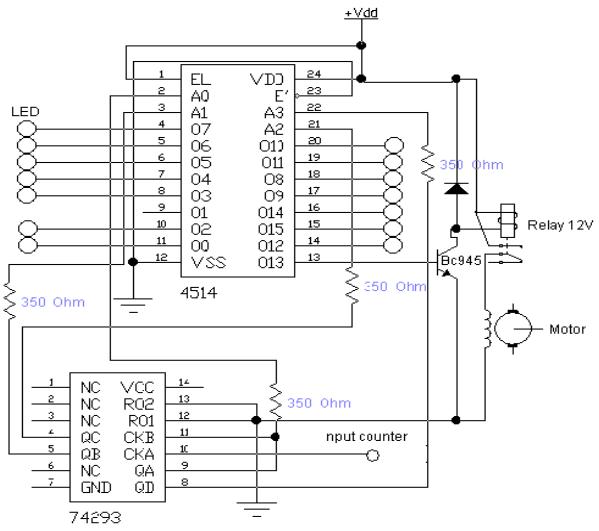

Figure 5: Schematic Diagram of Counter and 4-to-16 Line Decoder

No	Motor	Rotation	Number of pulse
1	Gripper	Clockwise	3
		Anti - Clockwise	4
2	Tool	Clockwise	5
		Anti - Clockwise	6
3	Wrist	Clockwise	7
		Anti - Clockwise	8
4	Elbow	Clockwise	9
		Anti - Clockwise	10
5	Shoulder	Clockwise	11
		Anti - Clockwise	12
6	Base	Clockwise	13
		Anti - Clockwise	14

Table 2: Numbers of Pulses at Each Motor

3.3 PARALLEL PORT PIN ASSIGNMENTS

For this project, the parallel port is used as a medium to interface between the software and the hardware part. In the standard parallel port, there are three registers are found. The registers' are:

1) Data register
2) Status register
3) Control register

Data register is connected to Data lines, Control register is connected to control lines and Status register is connected to Status lines. These registers are virtually connected to the corresponding lines. Therefore, whatever the user write to these registers, it will appear in the corresponding lines as voltages. It was proven when we measure the signal with a multimeter. If there are any voltages connected to the parallel port, we can read it from these registers (with some restrictions) [4]. For an example, if the user write '1' to Data register, the line Data0 will be driven to +5v. Therefore, it can be programmatically to turn ON and OFF at any of the data lines and Control lines. Figure 6 show the parallel port pin assignments.

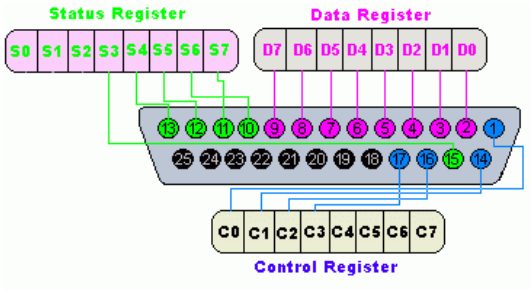

Figure 6: Parallel Port Assignment

The details of parallel port signal lines are given in table 2 below:

Pin No (DB25)	Signal name	Direction	Register - bit	Inverted
1	nStrobe	Out	Control-0	Yes
2	Data0	In/Out	Data-0	No
3	Data1	In/Out	Data-1	No
4	Data2	In/Out	Data-2	No
5	Data3	In/Out	Data-3	No
6	Data4	In/Out	Data-4	No
7	Data5	In/Out	Data-5	No
8	Data6	In/Out	Data-6	No
9	Data7	In/Out	Data-7	No
10	nAck	In	Status-6	No
11	Busy	In	Status-7	Yes
12	Paper-Out	In	Status-5	No
13	Select	In	Status-4	No
14	Linefeed	Out	Control-1	Yes
15	nError	In	Status-3	No
16	nInitialize	Out	Control-2	No
17	nSelect-Printer	Out	Control-3	Yes
18-25	Ground	-	-	-

Table 3: Parallel Port Pin Assignments

4.0 RESULT AND DISCUSSION

4.1 TESTING FOR THE WHOLE PROCESS

The whole process for this project has been developed and tested in the laboratory. After finishing some of the modifications of software and hardware that are required to achieve the objective, the result are obtained as shown below:

No of Test	Time taken to Accomplish the Whole Process (s)	Error
1	16.5	2.8
2	17.4	1.9
3	14.0	5.3
4	14.2	5.1
5	17.0	2.3
6	21.0	1.7
7	16.0	3.3
8	17.4	1.9
9	14.0	5.3
10	14.2	5.1
11	17.0	2.3
12	19.0	0.3
13	23.0	3.7
14	23.0	3.7
15	24.0	4.7
16	23.0	3.7
17	22.0	2.7
18	24.0	4.7
19	24.5	5.2
20	25.0	5.7
	Average of Whole Process = 19.3	Average of Error= 3.6

Table 4: Time Taken For the Whole Process and Error

*Error = Average of Whole Process - Time taken for each whole process

Refer to the Table 4, this test was taken for the whole process for the robotic arm to complete its movement. For this project, the robotic arm will pick up the desired object, move it to the desired place or angle and then returned to its original or initial state. The time was taken when the robotic arm begin to start to move until it stop. As we can see from Figure 7 and Figure 8, the time for the robotic arm to complete the whole process is not constant and inaccurate. Furthermore, the error also quite significant and it keeps rising after the 14th test. Therefore, in order to understand why this thing is happened, a few testing and troubleshooting was then be made. As a result, two major problems are found. The first problem is the gears that are allocated at the Base structure of this robotic arm. Actually, the Base will support all of its body weight and without knowing, this will lead to the misalignment gear and improve its friction when the Base is moving. When the base cannot move properly, it will lead to the time wasting and sometimes it could causes the Base cannot move to the desired location or angle.

The second problem is the harmonic current that produced when we connected the interface circuit with the load. The load consists of a few 6V dc motors and servomotors. The DC motor needed large amount of signal for it to operate and when the power is supplied to the DC motor, it will produce harmonic current and this sometimes, can cause the relay cannot accurately switched the selected motor.

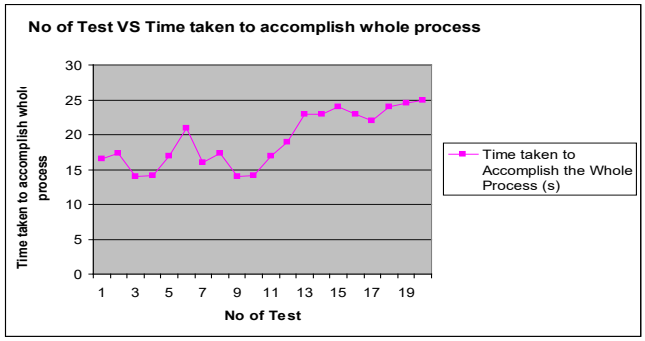

Figure 7: No of Test VS Time Taken to Accomplish

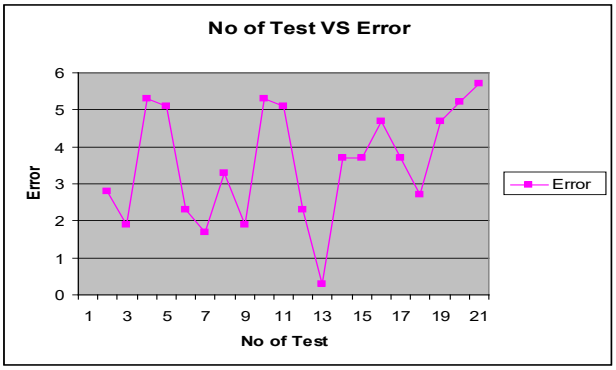

Figure 8: No of Test VS Error

4.2 TESTING THE GRIPPER

For this test, it is found that when the pulse signal is used to select a motor, the use of the output port for the parallel port can be reduced or saved. This is because; only one pin is used as a link to send a pulse signal into a circuit. Refer to Figure 9, to activate the gripper with anti-clockwise rotating, it need 4 pulses as a signal with 3.52 V as their amplitude. This amplitude is more than enough to give a signal to the counter and cause it to count as "1". Note that, the counter will give reading "1" when the voltage is between 2V to 6V; otherwise, the counter will give reading "0". This explanation also can be use to describe Figure 10.

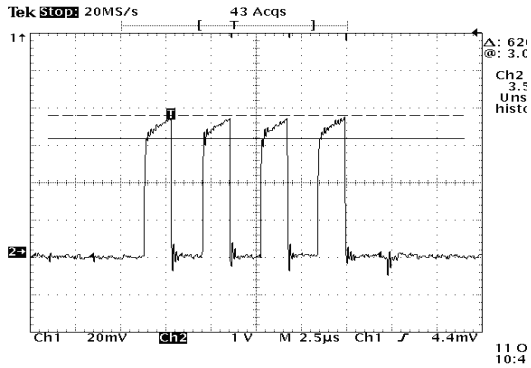

Figure 9: Number of Pulses to Select Gripper to Close.

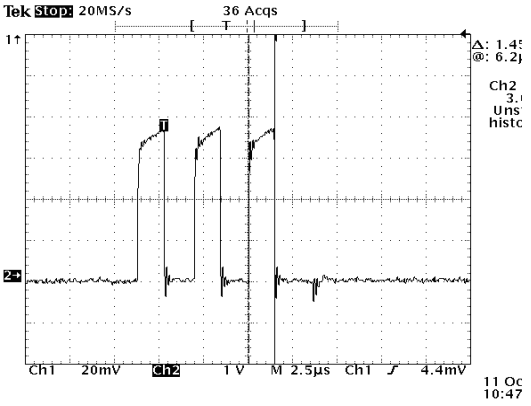

Figure 10: Number of Pulses to Select Gripper to Open.

Table 5 represents the data for the gripper. From this table, the difference rotation and their effect on a gripper movements were collected and this data is very important in order to find an equation that will be apply in visual basic software as a language to set up a width of opening and closing or to set up a degree. This is because, the servomotor that are used in this project do not have any sensor or feedback within it. So, in order to stop the movement for the servomotor, the timer is used based on the equations that are developed using this data. Setting of number is set in visual basic software.

Setting of number	Opening		Closing	
	Time (mS)	Width of opening (mm)	Time (mS)	Width of closing (mm)
10	40	1	30	76
20	70	3	80	73
40	150	9	170	63
60	230	14	240	59
80	310	21	310	52
100	380	32	380	48
120	460	40	470	46
140	540	49	540	38
160	620	53	620	32
180	690	65	690	28
200	780	76	780	23

Table 5: Opening and Closing Movement For the Gripper

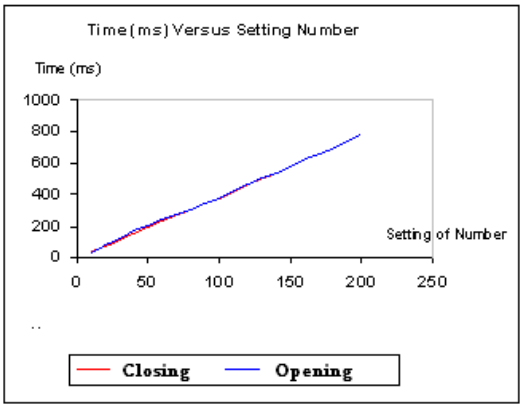

Figure 9: Graph Time (ms) Versus Setting Number

For the blue line (opening),

$y = mx + c$

$m = \frac{(620m - 70m)}{(180 - 20)} = 3.4375*10^{-3}$

$y = 3.4375*10^{-3}x + 5.625*10^{-3}$

$\mathbf{x = (y - 5.625*10^{-3}) / 3.4375*10^{-3}}$ ------- (1)

For the red line (closing),

$y = mx + c$

$m = \frac{(690m - 70m)}{(180 - 20)} = 3.875*10^{-3}$

$y = 3.875*10^{-3}x + 0.03$

$x = (y - 0.03) / 3.875*10^{-3}$ ------- (2)

After developing these two equations (equation 1 and 2), these equations will be applied in visual basic software as a language to find a width.

5.0 CONCLUSION

The VB programming that is developed in this project is actually a controller for the robotic arm. The method is that to interface the software with the interfacing circuit and feedback circuit through parallel port. In the other word, the parallel port will act as a medium between the computer, the circuitry and the robotic arm. By using the visual basic software, it is obviously and absolutely can increase the capability of the robotic arm.

For the robotic arm to complete the task that are given which is to pick an object, move the object to the desired location and to return back to its initial position requires several movement. These movements including the movement of base, shoulder, elbow, gripper and tool.

At the end of this project, the robotic arm and its software finally meet the objective needed.

6.0 FUTURE DEVELOPMENT

For the future development of this project, several modifications can be made in order to improve this project. The improvements are:

1. Add a few sensors to the gripper, elbow and wrist can make the robotic arm to move accurately. As an example, the proximity sensor can be added at the gripper and make the gripper to move with more precisely.

2. Furthermore, by inserting bearing into the structure of base and shoulder, it can make the movement of the robotic arm become smoother.

3. Finally, this project can be further develop by using wireless controller instead of using parallel port.

7.0 ACKNOWLEDGEMENTS

In the name of Allah S.W.T, the author would like to take this opportunity to express appreciation and gratitude to his project supervisors Pn Rosidah Bt Sam for her guidance towards this final project. The author also likes to thank family and individuals who have given encouragement to make this research successful.

8.0 REFERENCES

[1] David I. Schneider, *'An Introduction to Programming Using Visual Basic 6.0'*, Prentice Hall, 4th Edition, New Jersey, 1999

[2] Greg Perry, *Visual Basic in 12 Easy Lessons*, Sams Publishing, 1st Edition, Indianapolis, Indiana, 1995.

[3] Julia Case Bradley, Anita C. Millspaugh, '*Programming in Visual Basic 6.0*', Mc Graw Hill, 1999, pp1-549

[4] "*Parallel port interfacing made easy*" homepage, http://www.epanorama.net/circuits/parallel_output.html

[5] James L Fuller, *Robotics - Introduction, Programming and Projects*, 2nd edition, Prentice Hall, 2003.